



Bob, the Talking Skull Controlled by Jawduino!

I've been programming Halloween effects for a few years now, and recently my wife and I got a much bigger opportunity to put on a show. We're part of a Halloween prop building club, and the club had an exhibit at the big Maker Faire in San Mateo, CA. Our section of the exhibit had a mad scientist/steampunk/Victorian sci-fi theme.

During the fair, the event coordinator for the California Academy of Sciences (a science museum in Golden Gate Park, San Francisco) came by. The theme of our section impressed him, and he asked us if we would be willing to be a part of a one-night event. The Academy of Sciences hosts “Nightlife” — adults-only events every week — where you can chat with scientists about their work, have a drink, and enjoy the Academy without kids running around.

For Halloween, he wanted to do a “pop-up haunted house” with a combination of real science exhibits that let guests glove up and touch specimens, along with Halloween displays. The event would have DJs, people attending in costume, local professional actors in elaborate costumes and makeup, a big stage show as the finale, etc., and he saw our mad scientist props as fitting into the theme. He asked if we would like to be a part of it.

Of course, we said yes! We did it the first time in 2015; it went well, and we were invited back. So, now there was an opportunity to improve some of the effects. That just happens to bring us to the subject of this article: Bob, the talking skull.

What About Bob?

I’ve been trying out various setups to control a talking skull for a few years. Most of them work well, but it turns out the big pain in the neck is animating the jaw to sync with the dialog. We change what the skull says every year, and the software plays back six different MP3s in a random order. This means that each year, I write new dialog, get it recorded, convert it to a robot-sounding voice (more on that later), and then hand-animate the motions of the jaw to work with the soundtrack.

I’ve been looking for a way to make this process easier. One day on hauntforum.com, a user named mikkojay posted a video and code about his “Jawduino.” He had hacked a cheap audio meter and used it with an Arduino to control a talking skull’s jaw automatically — with no tedious hand animation needed!

The audio meter uses the KA2284: a chip designed to drive five LEDs for a VU or signal meter. Best part is an assembled board is available on eBay for about \$1 shipped!

The post about the Jawduino went up on August 14, 2016. The “Creatures of the NightLife” event would be held October 27, 2016. Since the parts would be coming from China, I ordered two of the little boards, so if one didn’t work, there was a backup. If they didn’t get here in time, well, I could use last year’s technology.

Not only was the clock ticking, but mikkojay’s original code was brief. It only used the Arduino to respond to the output of the audio meter and control the jaw. It didn’t control audio playback, and it didn’t have any way of

handling a list of MP3s.

As it turned out, the boards did make it in plenty of time. I added sound playback and set it up so it would cycle through six different MP3s, without repeating one until they were all played.

Everyone’s a Comic

With a prop like this, the main question is, how was it going to be used? We knew we would have a line of people at one end of Africa Hall, waiting to get into the haunted house area. Bob’s job would be to talk to the people waiting in line, cracking a few jokes in the process.

I had the lead actor of the theater group record six different voice tracks, and processed them into a robot-sounding voice using a tutorial on the Web. Why the extra effort to do a robot voice? It’s because the skull I have uses a servo to move the jaw, and it’s hard to disguise the sound!

Parting Ways

Instead of Bob being a “haunted” skull, we made him a cyborg. He started life as a kit from Cowlacious Designs, so he already had red LED eyes. Along the way, the original Cowlacious audio control board got damaged. Some chrome duct tape applied to his skull created the cyborg effect (**Figure 1**).

Here’s the list of materials we used to make the Jawduino controller:

- Arduino
- Screw Shield. Not totally necessary, but it makes life much easier (analog pins only)!
- KA2284 audio board
- Serial MP3 player. I use the Catalex unit, but there are lots of others available.
- Stereo mini phono plug to RCA jacks
- Servo extension cable
- Assorted jumper wires

Additionally, you will also need:

- A talking skull, with the jaw animated by a servo (not a DC hobby motor).
- Amplifier and speaker



Figure 1. A PVC stand is needed to hold the skull due to its weight. One rubber eyeball is removed to show the LED in the back of the socket.



Figure 2. KA2284 VU meter.

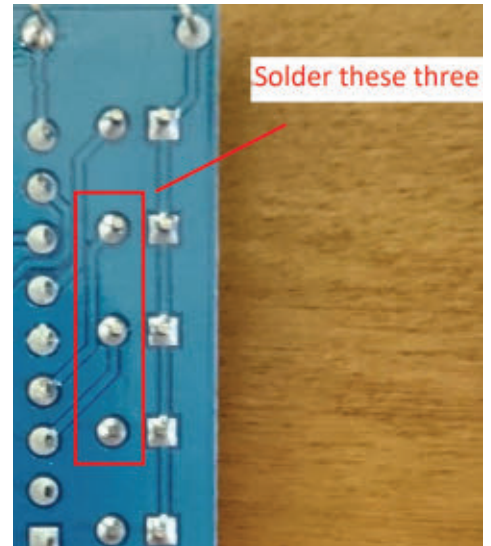
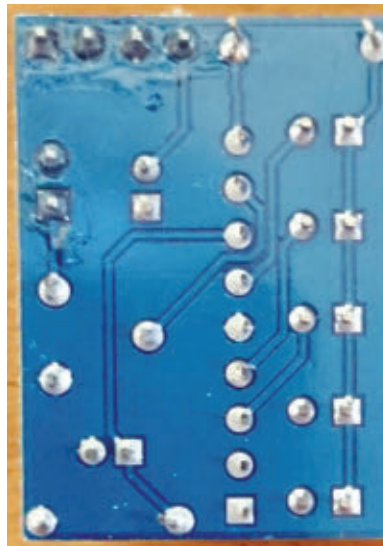


Figure 3. Solder to these connections.

Resources

Mikkojay's original web page on the Jawduino:
http://buttonbanger.com/?page_id=137

KA2284 audio board; easier to get than they were a year ago. Search Amazon and eBay for them.

UART control serial MP3 player — this is one listing:
www.dx.com/p/uart-control-serial-mp3-music-player-module-for-arduino-avr-arm-pic-blue-silver-342439#.WTx2_Ma1uUk

Arduino Screw Shield:
 Multiple vendors. I use one that comes in two parts, with servo headers on it. This gives you the most possible connections.

DriveSort:
www.anerty.net/software/file/DriveSort/?lang=en

I also used a piece of acrylic to mount everything, with standoffs. The unique component of this build is the KA2284 board (Figure 2).

Get the assembled version if you can. Thankfully, the available board follows the Samsung test circuit schematic. As the datasheet says, "The KA2284 is a monolithic integrated circuit designed for five-dot LED level meter drivers with a built-in rectifying amplifier."

Time for Assimilation

It's simple to use. Give it 3.5V-12V on VCC and GND, and connect one channel of your audio to IN and the second to GND. Adjust the pot to work with the level of the input, and you're in business; the LEDs will flash with the

music. What mikkojay figured out is that he could tap into the LEDs. Each time one of the LEDs lights up, you can detect it with the Arduino. For this project, he only used the middle three (Figure 3).

Solder three wires to the bottom of the board on the side of the LEDs that connect to the chip (not the side closest to the edge; these go to a resistor and then to VCC. Refer to the datasheet.) These three wires then go into three of the analog inputs on the Arduino.

Here's an example of how the angle of the servo is increased as each successive LED is turned on:

```
void audio_updates()
{
  audio_value = 0;
  if(analogRead(A0) < 341) audio_value += 60;
  if(analogRead(A1) < 341) audio_value += 60;
  if(analogRead(A2) < 341) audio_value += 60;
```

The angle of movement is being increased more and more as each successive LED is lit (e.g., the analog input voltage goes under a specific threshold). If all three middle LEDs are lit, then the maximum value hits 180. That is the maximum swing (in degrees) used for the servo.

Once you've soldered the wires to the bottom of the KA2284 board, then assembly is quick. I use the Screw Shield because it makes things much easier; it has multiple 5V and GND connections:

- KA2284 D2, D3, D4 to Arduino A0, A1, A2. VCC and GND to Arduino 5V, GND.
- MP3 audio board RX to Arduino A3. VCC and GND to Arduino 5V, GND.
- LED eyes to Arduino A4, GND, and pin.
- Jaw servo to Arduino A5. Use a servo cable.

Figure 4. Everything assembled and mounted.

After it was assembled, I mounted everything to a piece of acrylic as mentioned, and labeled all the connections (**Figure 4**). I don't want to have to figure out all these connections next year!

You need to customize a cable to trigger the KA2284. I used an old audio cable, and tapped the right channel to connect to the audio in and ground pins (**Figure 5**).

Setting Up the MP3s

One of the quirks of these small MP3 players is that they can't read the directory of a memory card and process it — you must organize the files on it in order. Name them like this:

- 001 Welcome.mp3
- 002 Scary rooms.mp3
- 003 Monsters ahead.mp3

Copy the files to the micro SD card; then, use a utility to sort the FAT on the card. This way, the commands from the Arduino will trigger the MP3 you want. The utility I use is DriveSort.

Finally, load the software onto the Arduino. There are two things you need to do to configure it for your setup. First, figure out the sweep of the servo that animates your skull's jaw. If you scroll all the way to the bottom of the code, the `set_minmax` function has comments in it that tell you what values to change for the motion you want. Ground pin 11 to get the servo to sweep through these values for testing.

Look at how long each MP3 is and change the `TalkTime` values at the top of the code to match.

The code is set up in a state machine, run by a timer. When a state is triggered, the LED eyes are turned on, and the command to play an MP3 is sent. The MP3 player — acting as a co-processor — will then play the file and stop on its own. The LED eyes are then turned off, based on how long the `TalkTime` is for each state.

The big insight I had while doing this was to put all the

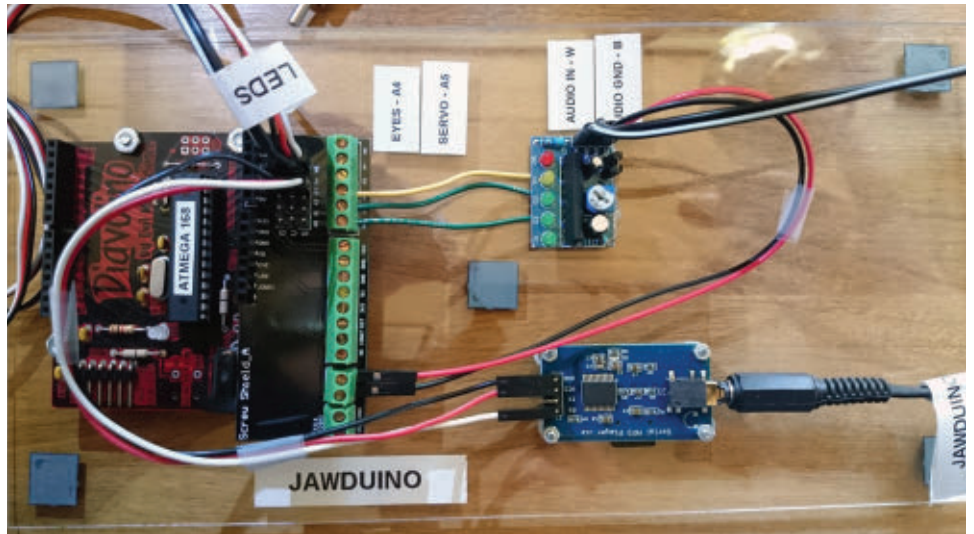


Figure 5. Hack an old audio cable to trigger the KA2284 board.

functions that monitor the audio and update the servos into the main loop, and only jump to the states briefly to trigger the MP3s. This lets the processor manage the servos until the timer cuts it off. You can change the pause between MP3s if you want; see the `WaitTime` variable at the top of the code.

In earlier iterations of Bob, I set the code up to randomly pick MP3s from a list until they were all used up, and then clear the list and start over. That turns out to be unnecessary — no one ever pays that much attention! They see the skull is talking, listen to one or two things it says, and move on. All I do now is run through the list, playing each track in order, then start over at the top.

Head Trip

Let's talk about the skull itself. Animating the jaw is straightforward; the servo is mounted in the top of the skull, and a wire or linkage goes down to the jaw. There are several approaches you can take to building this, depending on your budget and on how lifelike you want

Thanks to the following people who made this possible:

Mike "mikkojay" of HauntForum.com for the original concept and code.

Jon "JonnyMac" McPhalen for years of answering questions, awesome example code, and general inspiration.



Figure 6. The servo is mounted inside the skull, and moves the jaw with a length of piano wire.

Talking Skull Kit available from the *Nuts & Volts* Webstore!

If you'd like to build your own talking skull kit, you're in luck! To make it easier and convenient, the *Nuts & Volts* Webstore has stocked up with a variety of kits for differing skill levels (and parts bins) so you can build up your own animatronic! Go to store.nutsvolts.com for complete details on kit availability!

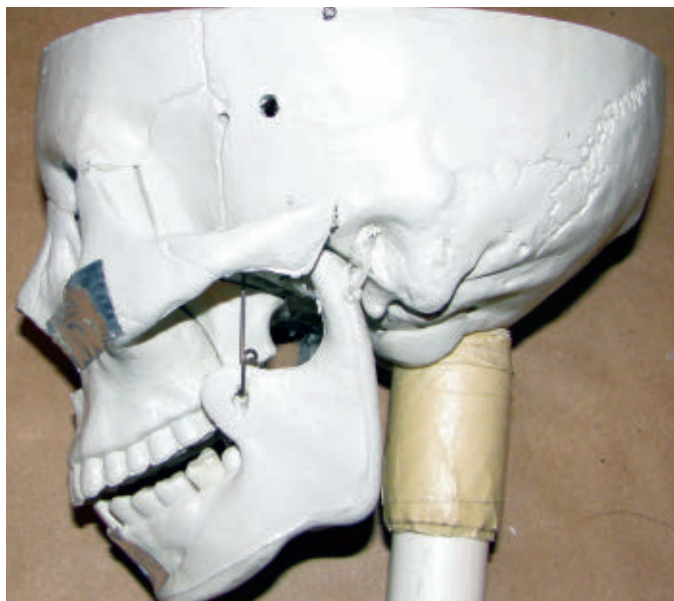


Figure 7. A hole is drilled in the jaw and the other end of the piano wire goes through it.



Figure 8. It looks technical, so it must mean something! The grill on the side is for the speaker.

the skull to look. The most realistic skulls are made for medical students and doctor's offices, and are anatomical lifesize skulls. They are made from very heavy-duty plastic, so you need a strong servo (**Figures 6** and **7**). Purchase the fourth-quality versions from skeletonstore.com, and see a great tutorial for installing a servo in one on Scary Terry's website at scary-terry.com. Another solution is to get a Lindberg Pirate Skull model kit. These are much lighter than the anatomical skulls, and allow for a lot of customization in the look, but have recently become harder to find.

However, if you can find a cheap skull with a moving jaw at the drugstore, that can work too! A small servo will be fine. You can use RC linkages or stiff piano wire to connect the servo to the jaw.

Mikkojay explains his budget approach on <http://buttonbanger.com>. Look for Jawduino along the top menu, and download his PDF.

It's All in the Presentation

Finally, there's the most important part of a prop: the presentation. Bob's head is a heavy lifelike skull, so I built a PVC stand to hold him steady. However, I needed an interesting way to hide it. I had part of an old Heathkit Breadboard Circuit Design Trainer ET-3100 that I had acquired. It was only the top part; somewhere along the way, it had been taken apart and all the innards were gone.

I got some black pebbled acrylic, made a box, and mounted the Heathkit to the front to act as a "control panel (**Figure 8**).” It needed something to make it look active.

Fortunately, I had an extra Velleman blinking LED kit in a drawer. I popped the LEDs into the holes on the front, and presto! Now, there were some blinking lights on the control panel.



Figure 11. Hey! Throw that away! I'm workin' here!

The box around Bob is big enough to hold the Jawduino assembly and a small audio amp and speaker. I mounted an old car speaker in a thick piece of corrugated cardboard and use a Tripath amp to drive it (**Figure 9**).

There are a couple of power cables that come out the back. Otherwise, it's a simple clean prop that can be set up on a table.

Once everything's connected (**Figure 10**), you power it up and it runs all night. Easy!

One unanticipated thing about having a prop like this at an event where people are drinking is that, well, people can be slob. Bob was set up on a table with plenty of extra room next to him. By the end of the night, people had dropped all their unwanted beer bottles and drink cups next to him (**Figure 11**).

However, Bob prevailed and will be entertaining folks again soon. **SV**



Figure 9. The speaker is from my long-gone 1995 Acura Integra.

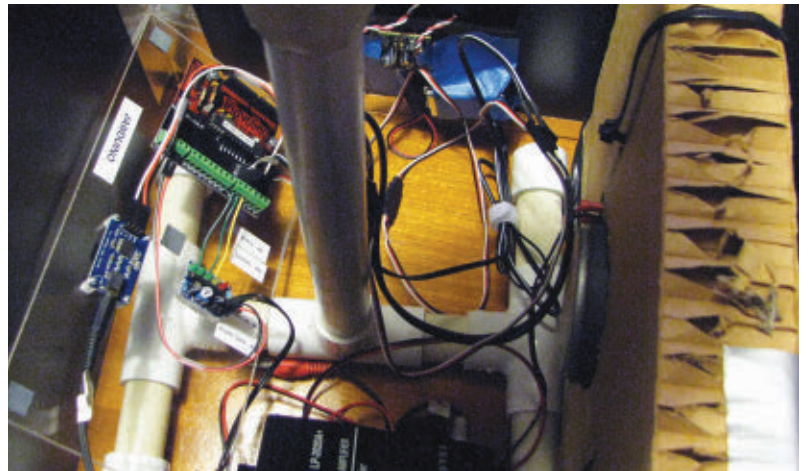


Figure 10. Everything drops inside the box. Then, power it up and put the cover on.



Make your machine *move*

MICRO LINEAR ACTUATORS

- 10mm-300mm stroke
- 25kg+ available force
- 6v-12v power supply
- 15g-100g net weight

ACTUONIX.COM